

(11) Japanese Patent Laid-Open No. 04-018857  
(43) Laid-Open Date: January 23, 1992  
(21) Application Number: 02-218214  
(22) Filing Date: August 21, 1990  
(32) Priority Date: April 23, 1990  
(31) Priority Number: 02-105352  
(71) Applicant: RICOH CO LTD  
(72) Inventor: MASAAKI ISHIKAWA

-----

1. Title of the Invention

IMAGE DATA COMPRESSION METHOD

2. Claims for the Patent

(1) An image data compression method for dividing input image data of one screen into a plurality of blocks, one block of  $p \times p$  pixels, and encoding and compressing each divided block in real time by an encoding method of combining an orthogonal transform and a variable length coding, including:

measuring the generated code amount for each block line for which the encoding is in progress, and controlling a code amount control parameter for each block line so that the code amount of one entire screen may be less than or equal to a predetermined target code amount, based on this measurement result.

(2) The image data compression method according to claim 1, characterized by including controlling the

code amount control parameter so that the target code amount at each point in time for each block line for which the encoding is in progress may be larger in the central part of said image, and smaller at the upper end part and the lower end part of said image.

(3) The image data compression method according to claim 1 or 2, characterized in that the code amount control parameter is a scale factor.

### 3. Detailed Description of the Invention

#### [ Industrial Application Field]

The present invention relates to an image data compression method for compressing and encoding the still image data for transmission or recording, including controlling the code amount after compression and encoding to be less than or equal to a required code amount.

#### [ Conventional Art]

As a highly efficient compression encoding technique for the natural image, a method of combining the orthogonal transform and the variable length coding has been examined. As an encoding method for the color natural image, the method of this kind will be adopted as the international standardization method (refer to Journal for The Institute of Image Electronics Engineers of Japan: Vol 18, No. 6, P398 to P407).

In the encoding method of this kind, the image

quality of decoded image and the code amount in encoding can be controlled by controlling a certain kind of parameter. The relationship between the image quality of decoded image and the code amount is such that as the code amount is larger, i.e., as the degree of compression is smaller, there is less degradation in the image quality from the original image, or conversely, as the code amount is smaller, i.e., the degree of compression is larger, there is more degradation in the image quality from the original image. However, in the encoding method of this kind, an adaptive process is performed using a local correlation of the image in encoding, whereby the highly efficient compression coding is enabled according to the image quality of object image.

[Problems to be Solved by the Invention]

By the way, generally, the image of minute pictorial pattern has a greater code amount, and the solid, flat image has a smaller code amount. Therefore, in the encoding method as described previously, if the encoding is performed using the same code amount control parameter, the obtained code amount is different for each object image. Accordingly, the code amount generated by encoding can not be predicted in advance, whereby it is not easy to encode any object image to obtain a certain code amount. Particularly, the application of the natural image (still image)

encoding may be made so far to a still image television telephone, a facsimile, storage in an image database, or the like, whereby it is not necessarily required to encode any object image to obtain a certain code amount.

However, if the encoding method of this kind is applied to the electronic still camera, a function capable of recording a certain number of images on a predetermined recording medium is required, and a technique for compressing and encoding each object image with a certain code amount is required. Especially in the consumer-oriented camera, the photographer needs to know how many more images can be recorded on the recording medium, whereby this technique is important. However, since the information amount contained in the image generally changes for each image as previously described, if all the object images are encoded with a definite code amount control parameter, there is inconvenience that the number of images recordable on the recording medium may change depending on the object images to be taken.

It is an object of this invention to provide an image data compression method for controlling so that the code amount after compression and encoding may be less than or equal to the code amount required for any object image.

[Means for Solving the Problems]

This invention provides an image data compression

method for dividing input image data of one screen into a plurality of blocks, one block of  $p \times p$  pixels, and encoding and compressing each divided block in real time by an encoding method of combining an orthogonal transform and a variable length coding, including measuring the generated code amount for each block line for which the encoding is in progress, and controlling a code amount control parameter for each block line so that the code amount of one entire screen may be less than or equal to a predetermined target code amount, based on this measurement result.

Also, the image data compression method includes controlling the code amount control parameter so that the target code amount at each point in time for each block line for which the encoding is in progress may be larger in the central part of the image, and smaller at the upper end part and the lower end part of the image.  
[Operation]

According to this invention, to make the code amount of one entire screen less than or equal to a predetermined target code amount, the average code amount up to that point in time of encoding is measured for each block line during the encoding, and the code amount control parameter is controlled so that the code amount may be less than or equal to the target average code amount up to the point in time of encoding.

That is, if the average code amount exceeds the

target average code amount, the code amount control parameter is controlled so that the code amount after compression is decreased for the next block line, or conversely if the average code amount is less than or equal to the target average code amount, the code amount control parameter is controlled so that the code amount after compression may be increased for the next block line. However, if the average code amount is less than or equal to the target average code amount and falls within a permissible range, the current code amount control parameter is maintained for the next block line.

In this case, the total target code amount may be changed halfway so that the target average code amount may be larger in the central part of the image, and smaller at the upper end part and the lower end part of the image, whereby more code amount is allocated to the central part to reduce a degraded image quality in the central part of decoded image.

[ Embodiments]

Figure 1 is a schematic diagram showing a procedure for an image data compression method according to one embodiment of the present invention.

This embodiment includes dividing input image data of one screen into a plurality of blocks, one block of  $p \times p$  pixels, for example,  $8 \times 8$  pixels (step T1), and performing a two-dimensional discrete cosine

transform (DCT) for each block (step T2). The DCT is a kind of orthogonal transform. Supposing that the transformation coefficient is  $F_{uv}$  ( $u, v = 0, 1, \dots, p-1$ ), and input image data for one block is  $f_{ij}$  ( $i, j = 0, 1, \dots, p-1$ ),  $F_{uv}$  is defined by [Equation 1]

$$\begin{aligned} \text{Where } C(w) &= 1/\sqrt{2} & (w = 0) \\ &= 1 & (w \neq 0) \end{aligned}$$

The obtained transformation coefficient  $F_{uv}$  indicates the component in which input image data of one block is decomposed into spatial frequencies. Among the transformation coefficients  $F_{uv}$ , the coefficient  $F_{00}$  indicates the value (DC component) proportional to the average value of input image data  $f_{ij}$  over  $p \times p$  pixels, in which as the variables  $u$  and  $v$  increase, the component (AC component) at higher spatial frequency is indicated.

The  $p \times p$  two-dimensional DCT coefficients  $F_{uv}$  thus obtained are divided by respective thresholds of a quantization matrix composed of  $p \times p$  thresholds, and quantized (step T3). Figures 2 and 3 show the examples of the quantization matrix for brightness signal and color difference signal. Each threshold of the quantization matrix is multiplied by the factor  $2^S$  ( $S = 0, \pm 1, \pm 2, \dots$ ) as will be described later, whereby the code amount and decoded image quality are controlled by changing the quantization step width.

The differences between the DC components of the quantized DCT coefficients and the DC components quantized in the previous block are taken (step T4). The number of bits for a difference is Huffman coded (step T5). The AC component is converted into one-dimensional number sequence by performing the zigzag scan within the block according to a zigzag scan table as shown in Figure 4 (step T6). The number of consecutive zero data is compressed by the run length coding (step T7). The two-dimensional Huffman coding is performed by the number of consecutive zero data and the number of bits in the effective coefficient obtained by the run-length coding (step T5). The Huffman coded data is stored as variable length coded data in the recording medium such as an IC card.

In this case, the code amount finally obtained as the variable length coded data can be controlled by changing each threshold of the quantization matrix in quantizing at step T3. In this example, each threshold of the quantization matrix is multiplied by the factor  $2^S$  and changed to control the code amount after compression and the quality of decoded image. Herein, the width  $S$  of the factor  $2^S$  is called a scale factor, and the relationship with the code amount is shown in a table at the following page.

[ Table 1 ]

That is, if the scale factor  $S$  is increased, the



quantization step width is larger, and the code amount is decreased. Conversely, if the scale factor  $S$  is reduced, the quantization step width is smaller, and the code amount is increased. This qualitative relationship holds without regard to the object image, but the quantitative relationship between scale factor  $S$  and data amount is not uniquely determined, whereby the code amount obtained for each object image is different even if the data is encoded with the same scale factor  $S$ .

Thus, in this embodiment, the average code amount up to the point in time of encoding is measured during the encoding, and the scale factor  $S$  is controlled so that the code amount is less than or equal to the target average code amount up to that point in time of encoding to perform a scale factor control (step T8). That is, when the average code amount exceeds the target average code amount, the quantization step width is made larger by increasing the scale factor  $S$ , so that the code amount after compression is reduced. Conversely, if the average code amount is less than or equal to the target average code amount and small beyond the permissible range, the quantization step width is made smaller by decreasing the scale factor  $S$ , so that the code amount after compression is increased. If the average code amount is less than or equal to the target average code amount and falls within the

permissible range, the current value of the scale factor  $S$  is maintained.

The scale factor control (step T8) is performed by dividing one screen composed of plural  $m \times n$  pixels as shown in Figure 5A into plural  $M \times N$  blocks, one block of  $p \times p$  pixels, as shown in Figure 5B, measuring the generated code amount for each obtained block column (hereinafter referred to as a block line) in the horizontal direction, and adjusting the scale factor  $S$  for the next block line based on the result.

The details of the scale factor process (step T8) will be described below in which each data is defined as follows.

- Number of pixels in one block:  $p \times p$
  - Number of blocks per block line:  $M$
  - Number of block lines:  $N$
  - Average code amount per target pixel:  $BR$
  - Average code amount permissible width per target pixel:  $BRW$
  - Target total code amount:  $TARCVOL$
- Where  $TARCVOL = BR \times M \times N \times p^2$
- Initial value of scale factor:  $S_0$
  - Scale factor control step width:  $\Delta S$
  - Generated code amount for each block line:

$CVOLBL(k)$

Where  $k$  is the variable indicating the block line number.

- Scale factor for each block line:  $S(k)$
- Block line number of block line where the encoding process is in progress: CURBN
- Block line number at which the scale factor is changed immediately before the point in time when the encoding process is in progress: BEFBN
- Accumulated code amount up to the point in time when the encoding process is in progress: CVOLAC(CURBN)

Where

[ Equation 2 ]

- Residual target average code amount at the point in time when the encoding process is in progress: TARBR(CURBN)

Where

[ Equation 3 ]

- Average code amount since the scale factor is changed immediately before up to the point in time when the encoding process is in progress: SMSFBR(CURBN)

Where

[ Equation 4 ]

The scale factor  $S$  is controlled under the following conditions (a) to (c), based on the above definitions.

- (a) If  $TARBR - BRW \leq SMSFBR \leq TARBR$ ,  $S(CURBN+1) = S(CURBN)$
- (b) If  $SMSFBR > TARBR$ ,  $S(CURBN+1) = S(CURBN) + \Delta S$
- (c) If  $SMSFBR < (TARBR - BRW)$ ,  $S(CURBN+1) =$

## S (CURBN) -AS

The condition (a) is the case where the average code amount is in an appropriate range of target value, in which the current scale factor value is maintained for the next block line.

The condition (b) is the case where the average code amount is beyond the permissible range and greater than the target value, in which the current scale factor value is increased by a certain step width for the next block line to decrease the generated code amount.

The condition (c) is the case where the average code amount is below the permissible range and smaller than the target value, in which the current scale factor value is decreased by the certain step width for the next block line to increase the generated code amount.

Figures 6A and 6B are the graphs showing the behavior of the generated code amount when such control is performed. The abscissa represents the block line number in the encoding process, and the ordinate represents the scale factor S on the upper part and the accumulated code amount CVOLAC on the lower part. However, the scale factor S has the reverse sign of positive or negative. Also, Figure 6A is a graph showing the behavior up to the end of encoding, and Figure 6B is a graph minutely showing the state during

the encoding. From the graph of Figure 6A, it can be found that the accumulated code amount for the entire image is controlled to be less than or equal to the target value.

The average code amount since the scale factor is changed immediately before up to the point in time when the encoding process is in progress is represented as an inclination of the straight line connecting the accumulated code amount (point a with ♦ sign in the graph) BEFBN at the point in time when the scale factor is changed immediately before and the accumulated code amount (point b with ◊ sign in the graph) CURBN at the point in time when the encoding process is in progress on the graph of Figure 6B. Also, the target average code amount at this point in time is represented as an inclination of the straight line connecting the accumulated code amount (point b with ◊ sign in the graph) at this point in time and the target accumulated code amount (point c with ▼ sign in the graph) for all the block lines.

Herein, the advantage of using the average code amount since the scale factor is changed immediately before up to the point in time when the encoding process is in progress as an observation amount for the control of the scale factor will be described below. For comparison, the other observation amounts for the control of the scale factor are given as follows.

(1) The total average code amount TOTALBR from the start of encoding up to the point in time when the encoding process is in progress,  
[Equation 5]

(2) The average code amount CURBR for only the block line at the point in time when the encoding process is in progress,  
[Equation 6]

In the case of using the total average code amount of (1) from the start of encoding up to the point in time when the encoding process is in progress, the behavior is less sensitive to a change in the generated code amount under the control of scale factor. This is shown in a graph of Figure 7. The average code amount of (1) is represented as an inclination of the straight line connecting the accumulated code amount (point a with ♦ sign in the graph) at the start point of encoding and the accumulated code amount (point d with ◊ sign in the graph) at the point in time when the encoding process is in progress on the graph of Figure 7. The graph of Figure 7 is the case where the scale factor at the point in time of starting the encoding is too small for the target code amount. But at the point in time c as indicated by ○ sign in the figure, though the average code amount for only this block line is already fully controlled for the target value, the total average code amount still exceeds the target

value, whereby the scale factor is changed. This appears as a local degradation in the image quality of the decoded image. In this manner, the excess control for the scale factor unfavorably has adverse influence on the image quality of the decoded image.

On the contrary, in the case of using the accumulated code amount since the scale factor is changed immediately before up to the point in time when the encoding process is in progress, the scale factor is controlled with an inclination of the straight line connecting the point b as indicated by • sign and the point c as indicated by ○ sign, whereby the excess control for the scale factor does not occur.

Next, in the case of using the average code amount of (2) for only the block line at the point in time when the encoding process is in progress, the behavior is conversely too sensitive to a change in the generated code amount of the image. This is shown in a graph of Figure 8. The average code amount of (2) is represented as an inclination of the straight line connecting the accumulated code amount (point c with ◊ sign in the graph) at the point in time when the encoding process is in progress and the accumulated code amount (point b with ♦ sign in the graph) for the immediately previous block line on the graph of Figure 8. Though the graph of Figure 8 is the case where the generated code amount locally increases during the

encoding, the average code amount for only this block line exceeds the target value at the point in time as indicated by  $\diamond$  sign in Figure 8, whereby the scale factor is changed. Of course, the property of the image changes with this block line as a boundary, in which if the generated code amount is increased over the long term, it is required to control the scale factor, but if the generated code amount happens to be increased for several block lines, the change of the scale factor may not be required. In such a case, if the average code amount of (2) for only the block line at the point in time when the encoding process is in progress is used, the scale factor is necessarily changed. This is the excess control for the scale factor, as in the previous case (1), and unfavorably appears as a local change in the image quality of the decoded image. If the average code amount since the scale factor is changed immediately before up to the point in time when the encoding process is in progress is used, the scale factor is controlled according to the inclination of the straight line connecting the point a as indicated by  $\bullet$  sign and the point c as indicated by  $\diamond$  sign in Figure 8, whereby the excess control for the scale factor does not occur.

Incidentally, in the ordinary photographed image, an important subject is often placed in the central part of the image. Therefore, the total target code



amount may be changed halfway so that the target average code amount is larger in the central part of the image and smaller at the upper end part and the lower end part of the image, thereby allocating a greater code amount to the central part.

Figure 9 shows an example in which the upper one-fourth of the image is the upper end part, the lower one-fourth of the image is the lower end part, and the central one-half is the central part, whereby the code amount that is  $(1+\alpha)$  times the total target code amount per pixel is allocated to the central part, and the code amount that is  $(1-\alpha)$  times the total target code amount per pixel is allocated to the upper end part and the lower end part. In this case, the target total code amount TARC VOL is defined as follows.

(a) When the block line number  $k$  is  $0 < k < N/4$   
(line a in Figure 9),

$$\text{TARC VOL} = \text{INITTARBR} \times \text{BR} \times (1-\alpha) \times M \times N \times p^2$$

(b) When the block line number  $k$  is  $N/4 < k < 3N/4$  (line b in Figure 9),

$$\text{TARC VOL} = \text{INITTARBR} \times \text{BR} \times (1+0.5\alpha) \times M \times N \times p^2$$

(c) When the block line number  $k$  is  $3N/4 < k < N$   
(line c in Figure 9),

$$\text{TARC VOL} = \text{INITTARBR} \times \text{BR} \times M \times N \times p^2$$

Where INITTARBR is an initial value (bit/pel) of the average code amount per target pixel. According to this embodiment, the target average code amount is made

larger in the central part of the image than the upper and lower end parts of the image, whereby it is possible to reduce a degradation in the image quality of the decoded image in the central part, and compress the entire image to the target code amount or less.

In the above manner, the generated code amount is measured for each block line during the encoding, and compared with the average code amount since the code amount parameter is changed immediately before up to the point in time when the encoding process is in progress, and the compression ratio control parameter is adjusted every time, whereby the entire image can be compressed to the target code amount or less without causing a local degradation in the image quality of the decoded image.

The control of the code amount may be made not only by using the scale factor, but also by changing the quantization matrix itself.

#### [Advantages of the Invention]

With this invention, it is possible to provide an image data compression method for assuring the number of images that can be compressed and stored in a certain storage capacity without causing a local degradation in the image quality of the decoded image in encoding and storing a plurality of still images in real time by a combined method of orthogonal transform and variable length coding.

[ Brief Description of the Drawings]

Figure 1 is a schematic diagram showing a procedure for an image data compression method according to one embodiment of the present invention;

Figures 2 and 3 are tables showing two examples of a quantization matrix for brightness signal and color difference signal;

Figure 4 is a table showing an example of zigzag scan;

Figure 5 is a schematic screen view for explaining the operation of this invention; and

Figures 6 to 9 are graphs showing how the generated code amount changes.

Figure 1

- T1     Blocking
- T3     Quantization
- T4     DC component difference coding
- T5     Huffman coding
- T6     Zigzag scan
- T7     Run length coding
- T8     Scale factor control
- #1     Input image data
- #2     Quantization matrix
- #3     Scale factor
- #4     Compressed data
- #5     Procedure of this invention

Figure 2

- #1     Quantization matrix for brightness signal

Figure 3

- #1     Quantization matrix for color difference signal

Figure 4

- #1     Table for zigzag scan

Figure 5

- #1     One pixel
- #2     One line
- #3     m pixels

#4     n lines  
#5     Total screen  $m \times n$  pixels  
#6     One block ( $8 \times 8$  pixels)  
#7     One block line  
#8     M blocks  
#9     N block lines  
#10    Total screen  $M \times N$  blocks  
#11    Screen configuration view

Figure 6

#1     Block line number

Figure 7

#1     Block line number

Figure 8

#1     Block line number

Figure 9

#1     Block line number

⑨ 日本国特許庁(JP)

⑩ 特許出願公開

⑩ 公開特許公報(A) 平4-18857

⑪ Int. Cl.<sup>5</sup>

識別番号

庁内整理番号

⑫ 公開 平成4年(1992)1月23日

H 04 N 1/41

G 06 F 15/66

H 04 N 7/13

3 3 0

3 3 0

B

C

Z

8839-5C

8420-5L

8420-5L

6957-5C

審査請求 未請求 請求項の数 3 (全9頁)

⑬ 発明の名称 画像データ圧縮方式

⑭ 特 願 平2-218214

⑮ 出 願 平2(1990)8月21日

優先権主張 ⑯ 平2(1990)4月23日⑰ 日本(JP)⑱ 特願 平2-105352

⑲ 発 明 者 石 川 雅 朗 東京都大田区中馬込1丁目3番6号 株式会社リコー内

⑳ 出 願 人 株 式 会 社 リ コ ー 東京都大田区中馬込1丁目3番6号

㉑ 代 理 人 弁 理 士 滝 野 秀 雄 外1名

明 細 書

1. 発明の名称

画像データ圧縮方式

2. 特許請求の範囲

- (1) 一 一面分の入力画像データを、1ブロックp × p 画素の複数ブロックに分割し、この分割した各ブロック毎に直交変換と可変長符号化とを組み合わせた符号化方式によって実時間で符号化して圧縮する画像データ圧縮方式において、上記符号化進行中の各ブロックライン毎に発生符号量を測定し、この測定結果に基づいて上記一面全体の符号量が所定の目標符号量以下となるように上記各ブロックライン毎に符号量制御パラメータを制御することを特徴とする画像データ圧縮方式。
- (2) 上記符号化進行中の各ブロックライン毎の各時点における目標符号量を、上記画像の中央部では大きく、上記画像の上端部および下端部では小さくなるように上記符号量制御パラメータ

を制御することを特徴とする請求項1記載の画像データ圧縮方式。

- (3) 上記符号量制御パラメータはスケールファクタであることを特徴とする請求項1および2記載の画像データ圧縮方式。

3. 発明の詳細な説明

(産業上の利用分野)

この発明は静止画データを圧縮符号化して伝送または記録する際に、圧縮符号化後の符号量が要求される符号量以下となるように制御する画像データ圧縮方式に関する。

(従来の技術)

✓ 自然画像に対する高効率な圧縮符号化技術として、直交変換に可変長符号を組み合わせた方式が検討されている。カラー自然画像の符号化方式としても、この種の方式が国際標準化方式として採用される予定である(画像電子学会誌: Vol. 18, No. 6, P. 398 ~ P. 407 参照)。

この種の符号化方式では、ある種のパラメータを制御することによって復号画像の画質と符号化における符号量とを制御することが出来る。復号画像の画質と符号量との関係は、符号量が大いほど、すなわち圧縮の度合いが小さいほど原画像からの画質の劣化は少なく、逆に符号量が小さいほど、すなわち圧縮の度合いが大いほど原画像からの画質の劣化は多くなる。ただし、この種の符号化方式では、符号化に際して画像の局所的な相関を利用して適応的な処理を行っているため、対象画像の画質に応じた高効率な圧縮符号化が可能となっている。

#### 〔発明が解決しようとする課題〕

ところで、一般に絵柄の細かな画像は符号量が多く、ベタの多い平坦な画像は符号量が少ない。このため、前述した符号化方式において、同じ符号量制御パラメータを用いて符号化を行うと、得られる符号量は対象画像毎に異なってくる。従って、符号化により発生する符号量を、予め予測す

ることが出来ず、任意の対象画像に対して一定の符号量が得られるように符号化することは容易ではない。特に、これまでの自然画像（静止画）符号化の応用は、静止画テレビ電話、ファクシミリ、画像データベースへの蓄積等が考えられていたため、任意の対象画像に対して一定の符号量が得られるように符号化を行うことは必ずしも要求されていなかった。

ところが、この種の符号化方式を、電子スチルカメラに応用しようとするとき、所定の記録媒体に一定の枚数の画像が記録できる機能が要求され、対象画像毎に一定の符号量で圧縮符号化する技術が要求される。特に民生用のカメラでは、記録媒体にあつて何枚の画像が記録できるかを撮影者に知らせる必要があるため、この技術は重要である。しかし、前述したように一般に画像に含まれる情報は画像毎に変化するため、対象画像を全て一定の符号量制御パラメータで符号化してしまうと、撮影する対象画像によって記録媒体に記録できる記録枚数が増減するという不都合が生じる。

#### 〔作 用〕

この発明は、一面画全体の符号量が所定の目標符号量以下となるように、符号化の途中でブロックライン単位にその符号化時点までの平均符号量を測定し、その符号量がその符号化時点までの目標平均符号量以下となるように符号量制御パラメータを制御する。

すなわち、平均符号量が目標平均符号量を超えるときは、次のブロックラインに対しては圧縮後の符号量が減少するように符号量制御パラメータを制御し、逆に平均符号量が目標平均符号量以下のときは、次のブロックラインに対しては圧縮後の符号量が増加するように符号量制御パラメータを制御する。ただし、平均符号量が目標平均符号量以下でかつ許容範囲内であれば、次のブロックラインに対しては現在の符号量制御パラメータを維持する。

また、この場合、目標平均符号量を、画像の中央部では大きく、画像の上端部および下端部では小さくするように、全体の目標符号量を途中で変

この発明は圧縮符号化後の符号量が任意の対象画像毎に要求される符号量以下となるように制御する画像データ圧縮方式を提供することを目的とする。

#### 〔課題を解決するための手段〕

この発明は、一面画分の入力画像データを、1ブロック× $N$ 画素の複数のブロックに分割し、この分割した各ブロック毎に直交変換と可変長符号化とを組み合わせた符号化方式によって実時間で符号化して圧縮する画像データ圧縮方式において、符号化進行中の各ブロックライン毎に発生符号量を測定し、この測定結果に基づいて一面画全体の符号量が所定の目標符号量以下となるように各ブロックライン毎に符号量制御パラメータを制御する。

また、符号化進行中の各ブロックライン毎の各時点における目標符号量を、画像の中央部では大きく、画像の上端部および下端部では小さくするように上記符号量制御パラメータを制御する。

$$=1 \quad (w=0)$$

化させ、中央部に多くの符号量を割り振ることによって復号画像の中央部での画質の劣化を小さくするようにしてもよい。

#### 【実施例】

第1図はこの発明による画像データ圧縮方式の処理手順の一実施例を示す流れ図である。

この実施例は、一面画分の入力画像データを、1ブロック $p \times p$ 画素、例えば $8 \times 8$ 画素からなる複数のブロックに分割し(処理T1)、各ブロック毎に2次元の離散コサイン変換(DCT)を施す(処理T2)。DCTは直交変換の一種であり、変換係数を $F_{uv}$ ( $u, v=0, 1, \dots, p-1$ )、1ブロック分の入力画像データを $f_{ij}$ ( $i, j=0, 1, \dots, p-1$ )とすると、

$$F_{uv} = \frac{4C(u)C(v)}{p^2} \sum_{i=0}^{p-1} \sum_{j=0}^{p-1} f_{ij} \cos \frac{(2i+1)u\pi}{2p} \cos \frac{(2j+1)v\pi}{2p}$$

値し、 $C(w) = 1/\sqrt{2} \quad (w=0)$

で定規され、得られる変換係数 $F_{uv}$ は1ブロック分の入力画像データを空間周波数に分解した成分を示している。変換係数 $F_{uv}$ のうち、係数 $F_{00}$ は入力画像データ $f_{ij}$ の $p \times p$ 画素の平均値に比例した値(直流成分)を表しており、定数 $u, v$ が大きくなるにつれて空間周波数の高い成分(交流成分)を表す。

こうして得られた $p \times p$ 個の2次元DCT係数 $F_{uv}$ は、 $p \times p$ 個の画素からなる量子化マトリクスの各画素で除算され量子化される(処理T3)。第2図および第3図に輝度信号用および色差信号用の量子化マトリクスの例を示す。量子化マトリクスの各画素にはそれぞれ後述する係数 $2^s$ ( $S=0, \pm 1, \pm 2, \dots$ )が乗算され、量子化ステップ幅を定数として符号量および復号画質を制御する。

量子化したDCT係数のうち、直流成分は前のブロックで量子化した直流成分と差分を取り(処理T4)、差分のビット数をハフマン符号化する(処理T5)。交流成分は、第4図に示すツグザ

グスキヤンのテーブルによってブロック内でツグザグスキヤンを行い、一次元の数値に変換し(処理T6)、連続する零データの個数をランレングス符号化によって圧縮し(処理T7)、ランレングス符号化した連続する零データの個数と有係数データのビット数とで2次元ハフマン符号化を行う(処理T5)。ハフマン符号化したデータは可変長符号化データとしてICカード等の記録媒体に蓄積する。

この場合、可変長符号化データとして最終的に得られる符号量は、処理T3における量子化の際の量子化マトリクスの各画素を変化させることによって制御できる。この実施例では、量子化マトリクスの各画素にある係数 $2^s$ を乗算して各画素を変化させ、圧縮後の符号量および復号画質を制御している。ここで、係数 $2^s$ の $s$ はスケールファクタと称され、符号量との関係は次頁の表に示すようになる。

① スケールファクタ	-	0	+
② 量子化ステップ幅	小	中	大
③ 符号量	多	中	少

④  
⑤

すなわち、スケールファクタ $s$ を大きくすると量子化ステップ幅も大きくなって符号量が少なくなり、逆にスケールファクタ $s$ を小さくすると量子化ステップ幅も小さくなって符号量が多くなる。この定性的な関係は対象画像によらず成立するが、スケールファクタ $s$ とデータ量との定量的な関係は一意には定まらず、同じスケールファクタ $s$ で符号化しても対象画像毎に得られる符号量は異なってくる。

そこで、この実施例では、符号化の途中でその符号化時点までの平均符号量を算出し、その符号量がその符号化時点までの目標平均符号量以下となるようにスケールファクタ $s$ を制御するスケールファクタ制御(処理T8)を行うようにしている。すなわち、平均符号量が目標平均符号量を越



えているときは、スケールファクタSを大きくして量子化ステップ幅を大きくし、圧縮後の符号量を減少させるようにし、逆に平均符号量が目標平均符号量以下でかつ許容範囲を超えて小さいときは、スケールファクタSを小さくして量子化ステップ幅を小さくし、圧縮後の符号量を大きくするようにする。平均符号量が目標平均符号量以下でかつ許容範囲内であれば、現在のスケールファクタSの値を維持するようにする。

スケールファクタ制御(処理T8)は、第5図(a)に示す画面構成図において、複数 $m \times n$ 画素からなる一面面を、第5図(b)に示す1ブロック $p \times p$ 画素からなる複数 $M \times N$ ブロックに分割し、得られる水平方向のブロック列(以下、ブロックライン、という)単位に発生符号量を測定し、その結果に基づいて次のブロックライン用のスケールファクタSの調整を行う。

次に、各データを次のように定義してスケールファクタ制御(処理T8)の詳細について説明する。

- ・符号化処理進行時点での残りの目標平均符号量:  $TARBR(CURBN)$ 、但し、

$$TARCVOL = \frac{\sum_{k=1}^{CURBN} CVOLBL(k)}{(N - CURBN) \times M \times p}$$

$$TARBR(CURBN) =$$

- ・直前にスケールファクタを変更してから符号化処理進行時点までの平均符号量:  $SMSFBR(CURBN)$ 、但し、

$$SMSFBR(CURBN) = \frac{\sum_{k=1}^{CURBN} CVOLBL(k)}{(CURBN - BEFBN + 1) \times M \times p}$$

これらの定義に基づいてスケールファクタSを、次の条件(a)~(c)のように制御する。

- (a)  $TARBR - BRW \leq SMSFBR \leq TARBR$  であれば、 $S(CURBN+1) = S(CURBN)$  とし、
- (b)  $SMSFBR > TARBR$  であれば、 $S(CURBN+1) = S(CURBN) + \Delta S$  とし、
- (c)  $SMSFBR < (TARBR - BRW)$  であれば、 $S(CURBN+1) = S(CURBN) - \Delta S$  とする。

条件(a)は平均符号量が目標値に対して適当な範囲にある場合で、次のブロックラインに対しても

- ・1ブロックの画素数:  $p \times p$ 、
- ・1ブロックライン当たりのブロック数:  $M$ 、
- ・ブロックライン数:  $N$ 、
- ・目標画素当たりの平均符号量:  $BR$ 、
- ・目標画素当たりの平均符号量許容幅:  $BRW$ 、
- ・目標全符号量:  $TARCVOL$ 、  
但し、 $TARCVOL = BR \times M \times N \times p$
- ・スケールファクタ初期値:  $S_0$ 、
- ・スケールファクタ制御ステップ幅:  $\Delta S$ 、
- ・ブロックライン毎の発生符号量:  $CVOLBL(k)$ 、但し、 $k$ はブロックライン番号を示す変数である。
- ・ブロックライン毎のスケールファクタ:  $S(k)$ 、
- ・符号化処理進行中のブロックラインのブロックライン番号:  $CURBN$ 、
- ・符号化処理進行時点の直前でスケールファクタを変更したブロックライン番号:  $BEFBN$ 、
- ・符号化処理進行時点までの累積符号量:  $CVOLAC(CURBN)$ 、但し、  
 $CVOLAC(CURBN) = \sum_{k=1}^{CURBN} CVOLBL(k)$

現スケールファクタ値を維持する。

条件(a)は平均符号量が許容範囲を超えて目標値を上回っている場合で、次のブロックラインに対しては現スケールファクタ値をあるステップ幅だけ増加させて発生符号量を減少させる。

条件(b)は平均符号量が目標値を許容範囲を超えて下まわっている場合で、次のブロックラインに対しては現スケールファクタ値をあるステップ幅だけ減少させ、発生符号量を増加させる。

第6図(a)および(b)はこのような制御を行ったときの発生符号量の様子をグラフに示したもので、横軸は符号化処理ブロックライン番号、縦軸は上部がスケールファクタS、下部が累積符号量CVOLACである。ただし、スケールファクタSは正負逆に取ってある。また、図(a)は符号化終了まで行った様子のグラフであり、図(b)は符号化途中の状態を詳しく表すためのグラフである。図(a)のグラフから両面全体での累積符号量が目標値以下に制御される様子が分かる。

直前にスケールファクタを変更してから符号化

処理進行時点までの平均符号量は、図4のグラフ上で直前のスケールファクタ変更時点の累積符号量（グラフ中の◆印の点a）BEFBNと符号化処理進行時点の累積符号量（グラフ中の○印の点b）CURBNとを結んだ直線の傾きとして表される。また、この時点での目標平均符号量は、この時点の累積符号量（グラフ中の○印の点b）と全ブロックラインに対する目標累積符号量（グラフ中の▼印の点c）とを結んだ直線の傾きとして表される。

ここで、スケールファクタ制御用の観測量として、直前にスケールファクタを変更してから符号化処理進行時点までの平均符号量を用いる利点について述べる。比較のために、他のスケールファクタ制御用の観測量として、

- ① 符号化開始から符号化処理進行時点までのトータルでの平均符号量 TOTALBR、

$$\text{TOTALBR} = \frac{\sum_{k=1}^{CURBN} \text{CVOLBL}(k)}{\text{CURBN} \times M \times P^2}$$

- ② 符号化処理進行時点のブロックラインのみ

での平均符号量 CURBR、

$$\text{CURBR} = \frac{\text{CVOLBL}(\text{CURBN})}{M \times P^2}$$

を挙げて説明する。

①の符号化開始から符号化処理進行時点までのトータルでの平均符号量を用いた場合は、スケールファクタ制御による発生符号量の変化に対して鈍感になる。これを示したのが第7図のグラフである。①の平均符号量は第7図のグラフ上で符号化開始時点の累積符号量（グラフ中の◆印の点a）と符号化処理進行時点の累積符号量（グラフ中の○印の点d）とを結んだ直線の傾きとして表される。第7図のグラフは、符号化開始時点でのスケールファクタが目標符号量に対して小さすぎた場合であるが、図中○印で示した時点cにおいて、このブロックラインのみでの平均符号量は既に目標値に対して十分に制御されているにも係らず、トータルでの平均符号量はまだ目標値を超えているため、さらにスケールファクタが変更されている。これは複写画像においては、局所的画質の

劣化となって現れる。このように、スケールファクタの過制御は複写画像の画質に悪影響を与えるので好ましくない。

これに対し、直前にスケールファクタを変更してから符号化処理進行時点までの平均符号量を用いた場合は、図中◆印で示された点bと○印で示された点cとを結んだ直線の傾きによりスケールファクタを制御するので、こういったスケールファクタの過制御が起こらないことが分かる。

次に、②の符号化処理進行時点のブロックラインのみでの平均符号量を用いた場合は、逆に画像中の発生符号量の変化に対して敏感すぎる。これを示したのが第8図のグラフである。②の平均符号量は同図のグラフ上で符号化処理進行時点の累積符号量（グラフ中の○印の点c）と直前のブロックラインでの累積符号量（グラフ中の◆印の点b）とを結んだ直線の傾きとして表される。同図のグラフは符号化途中で局所的に発生符号量が増大した場合であるが、図中○印で示された時点において、このブロックラインのみでの平均符号量

は目標値を超えるため、スケールファクタが変更されてしまう。もちろん、このブロックラインを境に画像の性質が変わり、発生符号量が長期に渡り増大したままの場合は、スケールファクタの制御が必要であるが、たまたま数ブロックラインだけ発生符号量が増大するような場合は、スケールファクタの変更が必要でない場合もある。このような場合でも、②の符号化処理進行時点のブロックラインのみでの平均符号量を用いると、スケールファクタが必ず変更されてしまう。これは先の①の場合と同様にスケールファクタの過制御であり、複写画像に局所的な画質の変化となって表れるので好ましくない。直前にスケールファクタを変更してから符号化処理進行時点までの平均符号量を用いた場合は、図中◆印で示される点aと○印で示される点cとを結んだ直線の傾きによってスケールファクタを制御するので、こういったスケールファクタの過制御が起こらないことが分かる。

ところで、通常の撮影画像では、画像の中央部

特開平4-18857(6)

に重要な被写体が置かれることが多い。このため、目標平均符号量を、画像の中央部では大きく、画像の上端部および下端部では小さくなるように全体の目標符号量を途中で変化する、中央部に多くの符号量を割り振るようにしてもよい。

第9図は、画像の上部1/4を上端部、下部1/4を下端部、中央1/2を中央部とし、中央部には全体の目標画素当たりの符号量の $(1+\alpha)$ 倍の符号量を割り振り、上端部および下端部には全体の目標画素当たりの符号量の $(1-\alpha)$ 倍の符号量を割り振るようにした例について示している。この場合、目標全符号量TARCVOLは次のように定義される。

- (a) ブロックライン番号kが、 $0 < k < N/4$  のとき(第9図中のラインa)、

$$\text{TARCVOL} = \text{INITTABR} \times \text{BR} \times (1-\alpha) \times M \times N \times p^2$$

- (b) ブロックライン番号kが、 $N/4 < k < 3N/4$  のとき(第9図中のラインb)、

$$\text{TARCVOL} = \text{INITTABR} \times \text{BR}$$

ことによっても可能である。

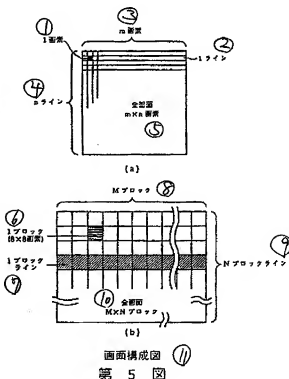
【発明の効果】

この発明によれば、複製機の静止画像を、直交変換と可変長符号とを組み合わせる方式により実時間で符号化して蓄積する際に、復号画像中に局所的な画質の劣化を生じることなく一定の記録容量中に圧縮蓄積可能な画像枚数が保証される画像データ圧縮方式を提供することが出来る。

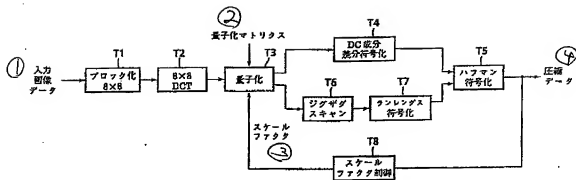
4. 図面の簡単な説明

- 第1図はこの発明による画像データ圧縮方式の処理手順の一実施例を示す概略図、  
第2図および第3図は輝度信号用および色差信号用の量子化マトリックスの例を示す表、  
第4図はジグザグスキャンの例を示す表、  
第5図はこの発明の動作を説明するための画面構成図、  
第6図乃至第9図は発生符号量の様子を示すグラフである。

- $\times (1+0.5\alpha) \times M \times N \times p^2$   
(c) ブロックライン番号kが、 $3N/4 < k < N$  のとき(第9図中のラインc)、
- $$\text{TARCVOL} = \text{INITTABR} \times \text{BR} \times M \times N \times p^2$$
- 但し、INITTABRは目標画素あたりの平均符号量初期値(bit/pel)である。この実施例によれば、目標平均符号量を、画像の上下端部に比べ画像の中央部で大きくとることによって復号画像の中央部の画質の劣化を小さくし、かつ画像全体の符号量を目標符号量以下に圧縮することができる。
- 以上のようにして、符号化の途中でブロックライン単位で発生符号量を測定し、直前に符号量パラメータを変更してから符号化処理進行時点までの平均符号量と比較してその度に圧縮率制御パラメータを制御することにより、復号画像に局所的な画質の劣化を招くことなく、画像全体の符号量を目標符号量以下に圧縮することが可能となる。なお、符号量の制御はスケールファクタを用いるほか、量子化マトリックスそのものを変化させる



画面構成図  
第5図



本案の処理手順 (5)  
第 1 図

Y/U	0	1	2	3	4	5	6	7
0	16	11	10	16	24	40	51	61
1	12	12	14	19	26	58	60	55
2	14	13	16	24	40	57	69	56
3	18	17	22	29	51	87	80	62
4	18	22	37	56	68	109	103	77
5	24	35	55	64	81	104	113	92
6	49	64	78	87	103	121	120	101
7	72	92	95	98	112	100	103	99

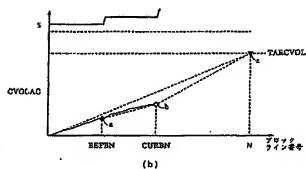
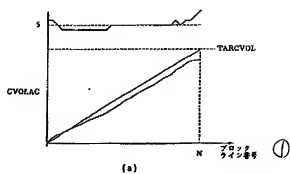
① 輝度信号の量子化マトリクス  
第 2 図

Y/U	0	1	2	3	4	5	6	7
0	17	18	24	47	66	99	99	99
1	18	21	26	66	99	99	99	99
2	24	26	56	99	99	99	99	99
3	47	66	99	99	99	99	99	99
4	99	99	99	99	99	99	99	99
5	99	99	99	99	99	99	99	99
6	99	99	99	99	99	99	99	99
7	99	99	99	99	99	99	99	99

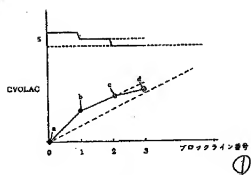
① 色差信号の量子化マトリクス  
第 3 図

Y/U	0	1	2	3	4	5	6	7
0	0	1	5	6	14	15	27	28
1	2	4	7	13	16	26	29	42
2	3	6	12	17	25	30	41	43
3	9	11	18	24	31	40	44	53
4	10	19	23	32	39	45	52	54
5	20	22	35	38	46	51	55	60
6	21	34	37	47	50	56	59	61
7	35	36	48	49	57	58	62	63

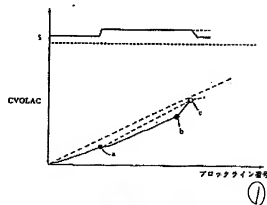
① ジグザグスキャンのテーブル  
第 4 図



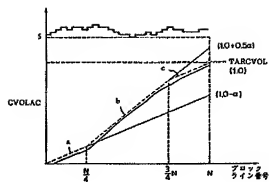
第 6 図



第 7 図



第 8 図



第 9 図

①